I Claim:

1. A method of applying instructions to a microprocessor during test mode, said method comprising:

a) entering a test mode establishing said microprocessor as a slave and a test controller as a master;

b) bypassing a first memory coupled to said microprocessor and forcing said microprocessor to execute instructions from an instruction queue; and

c) said test controller filling said instruction queue with instructions to be executed, said instructions originating from a test interface.

2. The method of Claim 1 wherein said test interface is serial.

3. The method of Claim 1 further comprising:

d) executing instructions from a second memory coupled to said microprocessor.

4. The method of Claim 3 wherein said second memory contains a set of pre-determined test instructions.

5. The method of Claim 4 further comprising:

e) switching execution between instructions in said instruction queue and said second memory.

6. The method of Claim 4 further comprising:

e) said test controller transferring an instruction to said instruction queue, said instruction causing said microprocessor to execute instructions from said second memory instead of said instruction queue.

7. The method of Claim 1 wherein said first memory is for holding instructions to be executed by said microprocessor when not in said test mode.

8. An architecture for applying instructions to a microprocessor during test mode, said architecture comprising:

a microprocessor coupled to a bus;

an instruction queue coupled to said microprocessor;

a test controller coupled to said bus, said test controller operable to load instructions received from a test interface into said instruction queue; and

a first memory coupled to said microprocessor, said first memory comprising pre-determined test instructions.

9. The architecture of Claim 8, further comprising:

a second memory comprising instructions to be run when not in said test mode, said architecture operable to bypass said second memory when in said test mode.

10. The architecture of Claim 9, wherein said test controller is not operable to access said second memory.

11. The architecture of Claim 9, wherein said second memory comprises a program flash memory.

12. The architecture of Claim 8, wherein said test controller is operable to force execution of instructions in said microprocessor between said instruction queue and said first memory by transferring a first instruction to said instruction queue.

13. The architecture of Claim 12, wherein said microprocessor is operable to cause a supervisory state to be entered upon receiving said first instruction from said instruction queue, wherein said microprocessor causes instructions from said first memory to be fed to said microprocessor.

14. The architecture of Claim 13, further comprising a register coupled to said bus, said register writeable in said supervisory state and not writeable when otherwise in said test mode, wherein said register is testable when in said supervisory state.

15. The architecture of Claim 8, wherein said microprocessor is further operable to allow said test controller to control the source of instructions for said microprocessor when a set of said pre-determined test instructions from said first memory have finished executing.